



19 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

12 Offenlegungsschrift
10 DE 197 17 167 A 1

51 Int. Cl.⁶:
G 06 F 15/163
G 06 F 9/46

21 Aktenzeichen: 197 17 167.2
22 Anmeldetag: 23. 4. 97
43 Offenlegungstag: 29. 10. 98

71 Anmelder:
International Business Machines Corporation,
Armonk, N.Y., US

74 Vertreter:
Kauffmann, W., Dipl.Phys. Dr., Pat.-Ass., 70569
Stuttgart

72 Erfinder:
Fehr, Clemens, 68163 Mannheim, DE; Riexinger,
Dieter, 75223 Niefern-Öschelbronn, DE

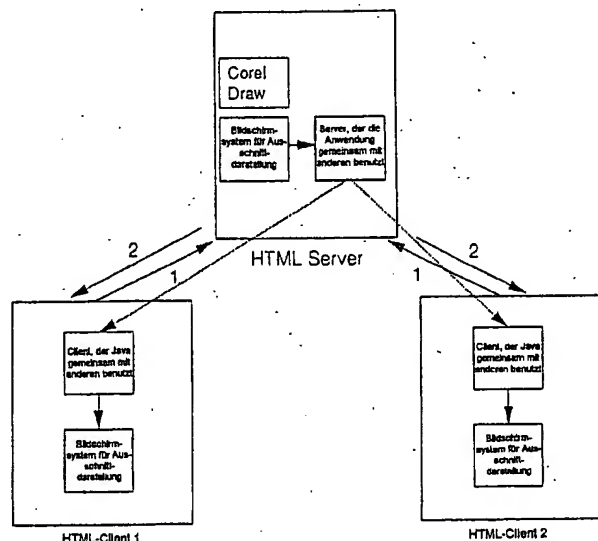
56 Entgegenhaltungen:
EP 04 75 581 A2
WO 96 29 664 A1
"HP SharedX: "in: HP Journal 4/94, 23-36;
"Vorsprung durch Technik" CT 4/97, 50-53;
"Network Computing" Elektronik 5/97, 122-126;

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

54 Webbrowser-basiertes Konferenzsystem

57 Beschrieben wird ein computergestütztes Application-Sharing Konferenzsystem, bei dem Konferenzteilnehmer gleichzeitig in einem Anwendungsprogramm arbeiten können. Insbesondere sind die jeweiligen Endsysteme über das Internet oder ein Intranet miteinander gekoppelt. Benutzereingaben und die Darstellung des Anwendungsprogramms auf dem jeweiligen Endsystem werden durch ein Sharing-Servermodul sowie ein Sharing-Clientmodul prozessiert. Diese Module werden den jeweiligen Konferenzteilnehmern beispielsweise als JAVA-Applets zur Verfügung gestellt. Das Konferenzsystem ist insbesondere Plattform-unabhängig.



BEST AVAILABLE COPY

DE 197 17 167 A 1

DE 197 17 167 A 1

Beschreibung

Die Erfindung betrifft computergestützte Video- oder Audiokonferenzsysteme, insbesondere sogenannte erweiterte Application-Sharing Systeme, bei denen zwei oder mehrere Gegenstellen direkt in einem Anwendungsprogramm arbeiten können.

Unter dem Begriff "Application-Sharing" verbergen sich im allgemeinen unterschiedliche Auffassungen. Im einfachsten Fall wird dem Anwender die Möglichkeit gegeben, ausgewählte Bildschirmhalte an die jeweilige Gegenstelle zu übertragen. Je nach Konzept wird dabei ein Videofenster benutzt, oder es werden zusätzliche Programmfenster auf der Gegenseite geöffnet. Beide Teilnehmer können so Daten oder Diagramme besprechen.

Beim sogenannten "erweiterten Application-Sharing" können nun beide Gegenstellen direkt in einem Anwendungsprogramm arbeiten. Das Programm läuft dabei nur auf einem Rechner, stellt aber Bedienungsfunktionen wie Tastatureingabe oder Maussteuerung auch auf dem bzw. den anderen an der Konferenz beteiligten Rechnern zur Verfügung. Kleinere Datenmengen lassen sich dabei direkt über die in Windows-basierten Systemen ("Windows" ist eine eingetragene Marke der Microsoft Corp.) bekannten Zwischenablagen austauschen. Dabei werden auf dem Ursprungsrechner die entsprechenden Daten markiert und in die Zwischenablage kopiert. Das Application-Sharing Modul sorgt dann dafür, daß die Daten in die Zwischenablage des Zielrechners übertragen werden, von wo aus sie beispielsweise über den "Einfügen"-Befehl in fast jedes beliebige Programm übernommen werden können.

Eine weitere Hilfe für computerübergreifende Teamarbeit stellt ein Notizblock ("Shared Whiteboard") wie beispielsweise in dem von der Firma Intel entwickelten Application-Sharing Programm "ProShare" ("ProShare" ist eingetragene Marke der Intel Corp.) dar. Es erlaubt die gemeinsame Bearbeitung eines Schmierzettels mit beliebig zu platzierenden Textblöcken, Markierungen und einfachen grafischen Skizzen. Darüber hinaus bietet ProShare die Möglichkeit, Einzelbilder aus dem angezeigten Videofenster einzufrieren und als Bitmap-Datei auf einer Festplatte zu speichern. So läßt sich leicht ein Paßfoto der Gesprächspartner oder ein Schnappschuß der vor eine Kamera gehaltenen Dokumente auf die Festplatte bannen.

Der Videokonferenzrechner ist in nahezu allen Fällen ein Windows-Rechner. Andere Betriebssysteme wie OS/2, Unix oder Mac (diese Namen sind eingetragene Marken jeweils der Firmen IBM, AT&T und Apple) werden derzeit kaum bis gar nicht unterstützt. Traditionelle Konferenzanwendungen erfordern deshalb, daß jeder Konferenzteilnehmer eine Lizenz der Konferenzanwendung erwirbt, wobei diese Anwendungen oft an eine einzige Betriebssystemplattform gebunden sind. Bisherige Application-Sharing Systeme sind daher als traditionelle Anwendungen konzipiert, deren Einsatz auf ein oder mehrere Betriebssysteme beschränkt ist. Beispiele solcher Anwendungen sind P2P, Intel ProShare oder Shared_X ("P2P" und "Shared_X" sind eingetragene Marken der IBM und HP). Die Benutzer einer Konferenzanwendung sind daher gezwungen, sich auf ein Produkt zu einigen oder Produkte zu verwenden, die einen gemeinsamen Standard benutzen. Konferenzanwendungen, die das gemeinsame Bearbeiten von Dokumenten erlauben, befolgen meist keinen gemeinsamen Standard.

Es sind darüber hinaus bereits Konferenz-Anwendungssysteme bekannt, die ein Sharing von HTTP-Dokumenten unter Benutzung von konventionellen WWW-(World Wide Web) Servern ermöglichen. Ein solches System ist in dem Artikel "A Design and Implementation of Conference Mosaic" der Autoren Euihyun Jung et al. als Conference Proceedings der "10th International Conference on Information Networking", POSTECH Information Research Lab, Pohang, South Korea, beschrieben. Dieses System weist eine gewöhnliche Benutzerschnittstelle, ähnlich dem bekannten "Mosaic" WWW-Browser, auf. Insbesondere basiert das System sogar auf dem Mosaic-Browser, der für die Konferenzanwendung entsprechend modifiziert wurde. In einem speziellen Arbeitsmodus werden die gemeinsam benutzten (shared) Daten vor einer Konferenz auf das jeweilige lokale System jedes Konferenzteilnehmers heruntergeladen, ohne daß eine Verbindung zu dem WWW-Server aufgebaut werden muß. Der Anwendungsbereich dieses Konferenzsystems reicht von interaktiven online-Spielen und gleichzeitig von mehreren Benutzern bearbeiteten Anwendungsprogrammen bis zum Sharing von Internet Dienstleistungen, wie z. B. der bekannte FTP (File Transfer Protocol) Gopher.

Ein weiteres Application-Sharing System, unter Verwendung des Internet, ist aus einem Artikel von H. M. Abdel-Wahab et al. mit dem Titel "XTV: A Framework for Sharing X Window Clients in Remote Synchronous Collaboration", veröffentlicht in IEEE, New York, 1991, Seiten 159-167, bekannt. XTV ist ein verteiltes System, das ein synchrones Sharing von X Window-Anwendungen in einer Gruppe von entfernt angesiedelten Benutzern ermöglicht, wobei die Benutzer mittels Workstations, basierend auf einer X-Plattform, über das Internet miteinander in Verbindung stehen. Die Hauptkomponenten dieses Systems sind insbesondere in der Weise ausgelegt, daß sie in anderen Konferenzsystemen und Anwendungen wieder verwendbar sind.

Der vorliegenden Erfindung liegt somit die Aufgabe zugrunde, in einem Informations- oder Rechnernetz ein gattungsgemäßes Konferenzsystem bereitzustellen, das unabhängig von der zugrundeliegenden Betriebssystemplattform das gemeinsame Benutzen beliebiger Anwendungen ermöglicht. Insbesondere ist dabei die Verwendung eines herkömmlichen Internet-Browsers auf Seiten der Konferenzteilnehmer avisiert.

Diese Aufgabe wird bei der vorliegenden Erfindung dadurch gelöst, daß eine auf einer ersten Systemeinheit angeordnete erste Funktionseinheit zum Generieren von Ausgabeinformationen einer auf der ersten Systemeinheit ausgeführten Anwendung und zum Übermitteln der Ausgabeinformationen an die mindestens zweite Systemeinheit, sowie zum Empfangen und Verarbeiten von Benutzereingaben von der mindestens zweiten Systemeinheit und zum Einspielen der Benutzereingaben in die Anwendung, und mindestens eine auf der mindestens zweiten Systemeinheit angeordnete zweite Funktionseinheit zum Präsentieren der Ausgabeinformationen auf der mindestens zweiten Systemeinheit, und zum Übermitteln der Benutzereingaben an die erste Funktionseinheit vorgesehen sind. Als Systemeinheit kommen sämtliche digitalen Endsysteme in Betracht, die einen Speicher zum Speichern der erforderlichen Funktionseinheiten sowie Mittel zur Aufnahme von Benutzereingaben und zur Präsentation der zugrundeliegenden Anwendung aufweisen. Insbesondere kommen als Systemeinheit Arbeitsplatzrechner (PCs), Workstations, oder die derzeit viel diskutierten Netzcomputer in Betracht. In einer bevorzugten Ausführungsform repräsentieren dabei die Ausgabeinformationen die grafische Benutzeroberfläche der jeweiligen Anwendung. Es wird hervorgehoben, daß die Erfindung eine gemeinsame Bearbeitung von

kompletten Anwendungsprogrammen erlaubt, und nicht nur von einzelnen Dokumenten.

Da das vorgeschlagene Konferenzkonzept lediglich die allgemein verfügbaren Web-Browser erfordert, ist es aufgrund seiner Unabhängigkeit vom jeweiligen Übertragungsprotokoll auch völlig plattformunabhängig. Zudem sind keinerlei Modifikationen an den bestehenden Web-Browsern erforderlich. Die eigentliche Konferenzanwendung kann daher z. B. als JAVA-Applet(s) verteilt und durch eine JAVA-Konferenzanwendung auf dem Web-Server koordiniert werden.

In einer vorteilhaften Ausführung der Erfindung kann vorgesehen sein, daß mindestens eine mit der ersten und der mindestens zweiten Systemeinheit verbundene dritte Systemeinheit vorgesehen ist, die mindestens eine der beiden Funktionseinheiten zur Übertragung an mindestens eine der ersten und zweiten Systemeinheiten bereitstellt, oder daß mindestens eine mit der ersten und der mindestens zweiten Systemeinheit verbundene dritte Systemeinheit vorgesehen ist, die mindestens eine der beiden Funktionseinheiten zur direkten Ausführung bereitstellt. Somit ist es möglich, die für die Konferenz erforderlichen Funktionseinheiten erst kurz vor Beginn einer Konferenz auf den Rechner des jeweiligen Konferenzteilnehmers zu übertragen oder aber die Funktionseinheiten auf einer weiteren Systemeinheit direkt auszuführen. Daher sind Änderungen an den bestehenden Endsysteimen der Teilnehmer nicht erforderlich.

Die Erfindung kann zudem sowohl bei über ein Intranet oder über ein Internet gekoppelten Systemeinheiten angewendet werden.

In einer weiteren Ausgestaltung der Erfindung kann vorgesehen sein, daß als Übertragungsprotokoll zwischen den Systemeinheiten ein TCP/IP (Transmission Control Protocol/Internet Protocol)-gesichertes Verbindungsprotokoll vorgesehen ist. Damit wird das erfindungsgemäße Konferenzsystem plattformunabhängig.

In einem weiteren Ausführungsbeispiel ist mindestens eine der Funktionseinheiten als "Applet" (Minianwendung) auf einer als Web(HTTP)-Server ausgelegten dritten Systemeinheit bereitgestellt, um das Applet von dem Web-Server vor Konferenzbeginn in eine der Systemeinheiten der Konferenzteilnehmer herunter zu laden.

Die Applets können danach z. B. auf einer Webseite der Systemeinheiten ausgeführt werden. Als Entwicklungsumgebung für die Applets bietet sich wegen seiner Plattformunabhängigkeit insbesondere JAVA an.

Die Präsentation der Benutzeroberfläche der jeweiligen Anwendung kann über ein Standard-Grafikformat, wie z. B. GIF, erfolgen, denn diese Formate werden von bestehenden Browsern unterstützt.

Weiterhin wird die der Erfindung zugrundeliegende Aufgabe gelöst durch ein Verfahren, bei dem die Ausgabeinformationen einer auf der ersten Systemeinheit aufgeführten Anwendung generiert werden, die Ausgabeinformationen an die mindestens zweite Systemeinheit übermittelt werden, die Benutzereingaben von der mindestens zweiten Systemeinheit empfangen und verarbeitet werden, die Benutzereingaben in die Anwendung eingespielt werden, die Ausgabeinformationen auf der mindestens zweiten Systemeinheit präsentiert werden, sowie die Benutzereingaben an die erste Systemeinheit übermittelt werden.

Weitere vorteilhafte Ausgestaltungen der erfindungsgemäßen Einrichtung und dem Verfahren ergeben sich aus den Unteransprüchen.

Auf der Basis der erfindungsgemäßen Architektur lassen sich beliebige Anwendungen plattformübergreifend verteilen. Eine anwendungsbezogene Standardisierung, wie dies zum Beispiel im Standard T.120 für eine Whiteboard-Anwendung geschieht, ist hier nicht notwendig, da die Konferenzanwendung zentral für alle Teilnehmer auf einem HTTP-(Hypertext Transfer Protocol) Server mit der vorgeschlagenen Konferenzerverweiterung läuft. Jeder Teilnehmer benutzt einen standardgemäßen HTTP-Browser ohne jede Erweiterung. Damit stehen den Benutzern eines Internets/Intranets sofort alle bisherigen Anwendungen zur gemeinsamen Benutzung zur Verfügung, ohne daß für jeden Teilnehmer eine Benutzungslizenz für eine Konferenzanwendung, z. B. für Intel ProShare, erworben werden muß.

In einem bevorzugten Ausführungsbeispiel wird daher die seitens der Erfindung vorgeschlagene Konferenzanwendung auf einem HTTP-Konferenz-Server implementiert. Vor Beginn einer Konferenz meldet sich jeder Konferenzteilnehmer bei dem Konferenz-Server an. Für alle Konferenzteilnehmer können Zugangsberechtigungen unter Prüfung ihrer Internetadressen eingerichtet und zur Verfügung gestellt werden. Für Intranetanwendungen kann die Konferenzanwendung in vorteilhafter Weise als JAVA-Klasse angeboten werden.

Im übrigen ist zur Durchführung der Konferenz kein weiteres Anwendungsprogramm (Konferenzanwendung) bei den Konferenzteilnehmern erforderlich. Die mittlerweile auf nahezu jedem Personalcomputer (PC) oder jeder Workstation eingerichteten Web-Browser stellen die einzige Voraussetzung für die Implementierung der Erfindung dar.

Im folgenden Teil wird die Erfindung anhand von Ausführungsbeispielen eingehender beschrieben. Hierbei zeigen

Fig. 1 den Informationsfluß in einer bei der Erfindung zugrundegelegten Systemumgebung;

Fig. 2a ein erstes Ausführungsbeispiel für den Aufbau einer Web-Sharing Konferenz;

Fig. 2b ein weiteres Ausführungsbeispiel entsprechend Fig. 2a;

Fig. 3 eine auf JAVA basierende Benutzerschnittstelle;

Fig. 4 ein in Anlehnung an Fig. 3 weitergebildetes Ausführungsbeispiel der Erfindung.

In Fig. 1 ist eine typische Systemumgebung, hier basierend auf das Internet, zusammen mit der der vorliegenden Erfindung zugrundeliegenden Architektur anhand einer Schemazeichnung dargestellt. In dieser exemplarischen Systemumgebung ist ein HTTP-Server, der als Application-Sharing Server ausgelegt ist, über das Internet mit einer Reihe von Konferenzteilnehmern, d. h. HTTP-Clients, verbunden. Es ist hier weiter angenommen, daß die gemeinsam von den Teilnehmern benutzte Anwendung auf einem der HTTP-Clients gespeichert ist und dort originär ausgeführt wird. Dieser ist ebenfalls Teilnehmer der Konferenz. Die aktuellen Ausgabeinformationen der Anwendung werden nun über den HTTP-Server jeweils an die HTTP-Clients übermittelt, damit diese ebenfalls mit der Anwendung interaktiv operieren können.

Die meisten der existierenden Anwendungsprogramme sind nicht für die Benutzung in einer computerunterstützten Konferenz ausgelegt. Die Benutzeroberflächen dieser Programme erscheinen nur auf einem Bildschirm. Um die Benutzeroberfläche auf mehreren Rechnern mit jeweils unterschiedlicher Betriebssystem-Plattform darstellen zu können, müssen insbesondere die grafischen Ausgaben der Anwendungsprogramme abgefangen und entsprechend an die Konferenzteilnehmer verteilt werden. Als gemeinsames Austauschformat dient bei dem erfindungsgemäßen Konferenzsystem ein HTML-(Hypertext Markup Language) Format. Die Ausgaben der zu verteilenden Anwendung an einen der in Fig. 1 dar-

gestellten HTTP-Clients werden zunächst in das HTML-Format umgewandelt und an den Application-Sharing Server übergeben. Dieser Serverprozeß kann auf dem gleichen oder einem entfernten Rechner ausgeführt werden.

Die Umwandlung in das HTML-Format geschieht bei nicht-HTML Anwendungen über das GIF-Bildformat. Bei JAVA Anwendungen wird zur Leistungssteigerung des Konferenzsystems die Verteilung über die grafischen Ausgabebe-
 5 fehle an die JAVA-Bibliotheken realisiert.

Bei dem in Fig. 2a dargestellten Ausführungsbeispiel für den Aufbau einer Web-Sharing-Konferenz ist angenommen, daß zwei über das Internet verbundene Endsyste (Clients) miteinander mittels eines ebenfalls an das Internet angekoppelten Servers kommunizieren. Es wird an dieser Stelle allerdings hervorgehoben, daß die Erfindung bereits in einem Netzwerk anwendbar ist, in dem nur ein Client mit einem Server in Verbindung steht, und in denen ein Application-Sha-
 10 ring auch ohne eine zugrundeliegende Konferenz stattfindet. Beispielsweise kann daran gedacht sein, daß neueste Softwareprodukte zum Austesten von einem Server heruntergeladen werden, wobei die Originalversion weiterhin auf dem Server verbleibt und auch dort ausgeführt werden kann.

Bei dem in Fig. 2a gezeigten Netzwerk sind sowohl auf den beiden Endsyste (Clients) als auch dem Server Windows-basierte Betriebssysteme installiert. Das betreffende Anwendungsprogramm, hier das Zeichenprogramm "Corel Draw" der Firma Corel ("Corel Draw" ist eingetragene Marke der Firma Corel) ist lediglich auf dem Server installiert. Die erforderliche Betriebssoftware zur Unterstützung der Sharing-Konferenz stellt zum einen ein sogenanntes Sharing-Server Modul auf dem jeweiligen Server, zum anderen ein sogenanntes Sharing-Client Modul auf den jeweiligen Endgeräten (Clients) bereit. Die genannten Programmodule werden in vorteilhafter Weise in der plattformunabhängigen Entwicklungssprache
 15 JAVA implementiert. Diese Module werden nachfolgend noch eingehender beschrieben.

Sharing-Servermodul

Das Sharing-Servermodul hat insbesondere die Aufgabe, die grafischen Ausgaben der zu verteilenden Anwendung zu erfassen und die Benutzereingabe aller Konferenzteilnehmer zu serialisieren und der Anwendung so zu übergeben, als
 25 seien die Eingaben an der lokalen Arbeitsstation erfolgt. Die folgenden Schritte erläutern den Ablauf einer Konferenz:

1. Laden des Servermoduls

Das Servermodul wird dem Konferenzteilnehmer in Form eines JAVA-Applets zur Verfügung gestellt. Das JAVA-Applet bietet dem Benutzer Funktionen an, mit deren Hilfe er eine beliebige Anwendung zur Verteilung auswählen kann.
 30

2. Erfassen der grafischen Ausgabedaten

Am Konferenzbeginn erhält jeder Teilnehmer eine komplette Ausgabe der Anwendung in Form einer komprimierten Bitmap, z. B. im GIF-Format. Ab diesem Zeitpunkt werden nur die Änderungen, z. B. Textbereiche erfaßt und verteilt.

3. Behandlung der Benutzereingaben

Das Servermodul erhält die Benutzereingabe direkt von den beteiligten Clientmodulen. Diese Eingaben werden zunächst auf Gültigkeit überprüft, d. h. ob das entsprechende Clientmodul berechtigt war, Benutzereingaben anzunehmen. Die Eingabedaten werden über die System-Warteschlange des jeweiligen Fenstersystems der Anwendung zur Verfügung gestellt. Da im allgemeinen auch die lokalen Benutzereingaben über diese Warteschlange abgearbeitet
 40 werden, ist für die Anwendung das Einspielen der Benutzereingaben transparent.

Sharing-Clientmodul

Das Sharing-Clientmodul hat die Aufgabe, die vom Servermodul erhaltenen grafischen Daten in einem Webbrowser oder einem separaten Fenster darzustellen. Alle Benutzereingaben, die im Anwendungsbereich des Webbrowsers stattfinden werden im Gegenzug dem Servermodul übergeben. Die folgenden Schritte erläutern den Ablauf einer Konferenz:

1. Laden des Clientmoduls

Das Clientmodul wird dem Konferenzteilnehmer in Form eines JAVA-Applets zur Verfügung gestellt. Das JAVA-Applet bietet dem Benutzer Funktionen an, mit deren Hilfe er das Eingaberecht zu einer verteilten Anwendung erlangen kann.
 50

2. Darstellen der grafischen Ausgabedaten

Bei Konferenzbeginn stellt das Clientmodul die verteilte Anwendung im Webbrowser als Grafik dar. Ab diesem Zeitpunkt werden nur noch die Änderungen an dieser Grafik, z. B. das Zeichnen eines Kreises, nachvollzogen.
 55

3. Erfassen der Benutzereingaben

Das Clientmodul erhält sämtliche Benutzereingaben, die auf der Anwendung im Webbrowser erfolgen. Diese Benutzereingaben werden an das Servermodul weitergeleitet, falls der Benutzer zuvor das Eingaberecht angefordert und erhalten hatte.
 60

Zu Beginn einer Konferenz wählen sich die Teilnehmer der Konferenz z. B. an einem Application-Sharing HTTP-Server ein. Dieser Vorgang kann durch ein vor der Konferenz an die jeweiligen Teilnehmer verteiltes Paßwort geschützt werden. Danach erhält jeder Teilnehmer den aktuellen Zustand der Konferenz als eine HTML-Seite angezeigt. Diese HTML-Seite stellt die Benutzeroberfläche der zu verteilenden Anwendung dar. Jeder Teilnehmer hat nun die Möglichkeit, selbst dieses Programm zu bedienen. Die Benutzereingaben der Teilnehmer werden von dem als JAVA-Applet bereitgestellten Sharing-Client Modul aufgenommen und von diesem an das Sharing-Server Modul weitergeleitet. Dort werden die Eingaben mittels des Sharing-Server Moduls dem jeweiligen Anwendungsprogramm als Benutzereingaben eingespielt. Die entsprechenden Änderungen an der Benutzeroberfläche werden wiederum an alle Konferenzteilnehmer
 65

verteilt.

Bei dem in Fig. 2b dargestellten Ausführungsbeispiel ist, im Gegensatz zu der Ausführungsform gemäß Fig. 2a, das jeweilige Anwendungsprogramm nicht auf dem Server, sondern auf einem der Clients angeordnet. Dementsprechend ist auch das notwendige Sharing-Server Modul auf diesem Client installiert. Dieser Client verfügt zudem über ein HTTP-Server-Interface.

In einem weiteren Ausführungsbeispiel der Erfindung steht den Benutzern eine Reihe von Grafikoperationen zur Verfügung, die u. a. das Annotieren auf der verteilten Anwendung erlauben. Diese Funktionen können in vorteilhafter Weise ebenfalls als JAVA-Klassen auf dem jeweiligen Client-Rechner ausgeführt werden. Zur Verdeutlichung wird angenommen, daß in einer Architektur entsprechend Fig. 2a einer der Teilnehmer die Annotation "Dreieckzeichnen" ausführt. Das entsprechende JAVA-Applet sendet die Informationen über das gezeichnete Dreieck, z. B. Koordinaten, Farben oder Rand, als Rückmeldung an den Konferenz-HTTP-Server. Dieser verteilt die Informationen dann an alle Clients, die die Operation ebenfalls durch das gleiche JAVA-Applet ausführen lassen.

Es wird hervorgehoben, daß erst durch die neuen Fähigkeiten, die JAVA-Klassen in Internet- bzw. Intranet-Anwendungen zur Verfügung stellen, eine interaktive browserbasierte Konferenzanwendung möglich wird. Die auf den Client-Maschinen ausgeführten JAVA-Applets sind in der Lage, auf Ereignisse des Konferenz HTTP-Servers zu reagieren. Ein Benutzer der Konferenzanwendung muß nicht aktiv das Erneuern der jeweiligen HTML-Seite veranlassen. Diese Funktionalität der JAVA-Klassen erlaubt auch das Entwickeln von Diensten zur Synchronisation der Benutzereingaben.

Die Synchronisation der Benutzereingaben kann auf unterschiedliche Art erfolgen. Meist wird das Benutzerrecht zur Eingabe mittels eines Eingabe-Tokens geregelt. Der Besitzer eines Tokens entscheidet, an wen er das Eingaberecht weitergibt. Jeder Teilnehmer muß zuvor durch eine Funktion an der Benutzerschnittstelle das Eingaberecht anfordern.

Wie bereits beschrieben, ist es besonders vorteilhaft, die für die Konferenz erforderlichen Steuermodule in der Entwicklungssprache JAVA zu programmieren. In einer anderen Ausführungsform der Erfindung wird dieses Konzept dahingehend erweitert, daß auch die zugrundeliegenden Anwendungsprogramme in JAVA programmiert sind. Im folgenden Teil wird daher ein Verfahren beschrieben, um das Verteilen von beliebigen JAVA-Anwendungen zu realisieren.

Das Verfahren ersetzt die in JAVA standardmäßig enthaltene Komponente zur Darstellung und Behandlung von Elementen der Benutzerschnittstelle durch eine neue Komponente. Üblicherweise befindet sich die Implementierung der für das JAVA-Laufzeitsystem notwendigen Klassen in einer Datei namens "Classes.Zip". Dies trifft für alle Plattformen zu, für die es eine JAVA-Unterstützung gibt.

In Fig. 3 ist nun dargestellt, wie eine in JAVA generierte Benutzerschnittstelle durch JAVA-Klassen definiert ist, die in den genannten Zip-Dateien abgelegt sind. Diese Datei liegt für jeden Client vor, wobei die Schnittstelle zu den Klassen für alle Plattformen identisch ist. Demgegenüber stellt sich jedoch die Implementierung der Klassen auf den verschiedenen Plattformen unterschiedlich dar, z. B. werden zur Darstellung eines Schaltknopfes ("Button") in JAVA-Programmen Objekte der Klasse "Button class" erzeugt. Die Implementierung dieser Klasse ist auf jeder Plattform unterschiedlich und Bestandteil des jeweiligen JAVA-Laufzeitsystems. Zur Darstellung eines Buttons werden die auf der jeweiligen Plattform vorhandenen originären Grafikmechanismen ausgenutzt.

Die Architektur des JAVA-Laufzeitsystems ist bis auf die Schnittstellen, die in der genannten Zip-Datei enthalten sind, durchgehend identisch. Erst innerhalb dieser Datei befinden sich die plattformabhängigen Implementierungen. Diese Tatsache wird bei der vorliegenden Erfindung dadurch ausgenutzt, daß die Implementierung der dem jeweiligen Benutzer zugeordneten Klassen ersetzt wird durch eine Verteilkomponente. Diese Verteilkomponente sorgt für den Austausch der zur Verteilung der Anwendung notwendigen Daten. Um die Aufgabe der Darstellung und Behandlung von Elementen der Benutzerschnittstelle erfüllen zu können, wird die bestehende Implementierung in die Verteilkomponente eingebettet und von dieser mitbenutzt.

Dies wird nun anhand von Fig. 4 verdeutlicht: Der Prozeß der Ersetzung von Teilen des bestehenden Programmcodes in der genannten Zip-Datei durch einen neuen Teil ist auf allen Systemen übereinstimmend, da das Dateiformat und das Format der Anweisungen innerhalb der Datei standardisiert sind. In Fig. 4 ist dies am Beispiel der Ersetzung des Programmcodes für die Behandlung von einfachen Schaltflächen (Buttons) dargestellt.

Als weiteres Beispiel wird nun anhand eines Pseudo-Codes die Verteilung der Methode zum Setzen der Beschriftung eines Buttons beschrieben:

```
public class Button {  
.  
.  
.  
public void set Label (string text)  
// oldButton ist ein Objekt der ursprünglichen Button-  
// Klasse.  
oldButton.set Label (text),  
// ruft den ursprünglichen  
// Code zum Setzen der Beschriftung auf;  
// die angeschlossenen Clients über die Änderung der  
// Beschriftung informieren,  
verteile Aufruf ("Button.set Text", text oldButton)  
}
```

Das beschriebene Verfahren erfüllt somit höchste Ansprüche an die Verarbeitungsgeschwindigkeit, da keine Bilddaten über das Netzwerk ausgetauscht werden müssen, sondern nur Informationen wie z. B. "Button" wurde gedrückt oder "Button" wird an Position (XY) erzeugt. Auf der jeweiligen Client-Maschine werden diese Befehle wieder in Aufrufe an die entsprechenden Klassen für Schaltflächen in der genannten Datei "Classes.Zip" umgewandelt.

1. Einrichtung zum gemeinsamen Bearbeiten einer Anwendung auf einer ersten digitalen Systemeinheit und auf mindestens einer mit der ersten Systemeinheit verbundenen zweiten digitalen Systemeinheit, **gekennzeichnet durch**
 - eine auf der ersten Systemeinheit angeordnete erste Funktionseinheit zum Generieren von Ausgabeinformationen einer auf der ersten Systemeinheit ausgeführten Anwendung und zum Übermitteln der Ausgabeinformationen an die mindestens zweite Systemeinheit, sowie zum Empfangen und Verarbeiten von Benutzereingaben von der mindestens zweiten Systemeinheit und zum Einspielen der Benutzereingaben in die Anwendung;
 - mindestens eine auf der mindestens zweiten Systemeinheit angeordnete zweite Funktionseinheit zum Präsentieren der Ausgabeinformationen auf der mindestens zweiten Systemeinheit, und zum Übermitteln der Benutzereingaben an die erste Funktionseinheit.
2. Einrichtung nach Anspruch 1, dadurch gekennzeichnet, daß mindestens eine mit der ersten und der mindestens zweiten Systemeinheit verbundene dritte Systemeinheit vorgesehen ist, die mindestens eine der Funktionseinheiten zur Übertragung an mindestens eine der ersten und zweiten Systemeinheiten bereitstellt.
3. Einrichtung nach Anspruch 1, dadurch gekennzeichnet, daß mindestens eine mit der ersten und der mindestens zweiten Systemeinheit verbundene dritte Systemeinheit vorgesehen ist, die mindestens eine der Funktionseinheiten zur direkten Ausführung bereitstellt.
4. Einrichtung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die Systemeinheiten über ein Intranet oder Internet miteinander verbunden sind.
5. Einrichtung nach Anspruch 4, dadurch gekennzeichnet, daß als Übertragungsprotokoll zwischen den Systemeinheiten ein TCP/IP-Protokoll vorgesehen ist.
6. Einrichtung nach einem der Ansprüche 2, 4 oder 5, dadurch gekennzeichnet, daß mindestens eine der Funktionseinheiten als Applet auf einer als Web-(HTTP-)Server ausgelegten dritten Systemeinheit bereitgestellt ist, um das Applet von dem Web-Server in eine der Systemeinheiten herunter zu laden.
7. Einrichtung nach Anspruch 6, dadurch gekennzeichnet, daß das Applet auf einer Webseite mindestens einer der ersten oder zweiten Systemeinheiten ausgeführt wird.
8. Einrichtung nach einem der Ansprüche 2 oder 4 bis 7, dadurch gekennzeichnet, daß mindestens eine der Funktionseinheiten als JAVA Applet ausgelegt ist.
9. Einrichtung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die Ausgabeinformationen die grafische Benutzeroberfläche der Anwendung repräsentieren.
10. Einrichtung nach Anspruch 9, dadurch gekennzeichnet, daß die Benutzeroberfläche im Bitmap- oder GIF-Format repräsentiert ist.
11. Einrichtung nach einem der Ansprüche 2 oder 4 bis 10, gekennzeichnet durch eine Präsentation der Ausgabeinformationen mittels eines bei mindestens einer der Systemeinheiten vorliegenden Web-Browsers.
12. Einrichtung nach einem der Ansprüche 2 oder 4 bis 11, gekennzeichnet durch Mittel zum Anmelden einer gemeinsamen Bearbeitung einer Anwendung bei der dritten Systemeinheit.
13. Verfahren zum gemeinsamen Bearbeiten einer Anwendung auf einer ersten digitalen Systemeinheit und auf mindestens einer mit der ersten Systemeinheit verbundenen zweiten digitalen Systemeinheit, dadurch gekennzeichnet, daß
 - Ausgabeinformationen einer auf der ersten Systemeinheit ausgeführten Anwendung generiert werden,
 - die Ausgabeinformationen an die mindestens zweite Systemeinheit übermittelt werden,
 - Benutzereingaben von der mindestens zweiten Systemeinheit empfangen und verarbeitet werden,
 - die Benutzereingaben in die Anwendung eingespielt werden,
 - die Ausgabeinformationen auf der mindestens zweiten Systemeinheit präsentiert werden, und
 - die Benutzereingaben an die erste Systemeinheit übermittelt werden.
14. Verfahren nach Anspruch 13, dadurch gekennzeichnet, daß die Verarbeitung der Ausgabeinformationen und der Benutzereingaben mittels mindestens einer auf einer der Systemeinheiten vorliegenden Funktionseinheit erfolgt, wobei mindestens eine der Funktionseinheiten von einer mit der ersten und der mindestens zweiten Systemeinheit verbundene dritte Systemeinheit bereitgestellt und an mindestens eine der ersten und zweiten Systemeinheiten übertragen wird.
15. Verfahren nach Anspruch 13, dadurch gekennzeichnet, daß die Verarbeitung der Ausgabeinformationen und der Benutzereingaben mittels mindestens einer auf einer der Systemeinheiten vorliegenden Funktionseinheit erfolgt, wobei mindestens eine der Funktionseinheiten zur direkten Ausführung auf einer mit der ersten und der mindestens zweiten Systemeinheit verbundene dritte Systemeinheit bereitgestellt wird.
16. Verfahren nach einem der Ansprüche 13 bis 15, dadurch gekennzeichnet, daß die Systemeinheiten über ein Intranet oder Internet miteinander verbunden werden.
17. Verfahren nach Anspruch 16, dadurch gekennzeichnet, daß Informationen zwischen den Systemeinheiten mittels eines TCP/IP-Protokolls übertragen werden.
18. Verfahren nach Anspruch 15 oder 16, dadurch gekennzeichnet, daß mindestens eine der Funktionseinheiten als Applet auf einer als Web-(HTTP-)Server ausgelegten dritten Systemeinheit bereitgestellt wird, um das Applet von dem Web-Server in eine der Systemeinheiten herunter zu laden.
19. Verfahren nach Anspruch 18, dadurch gekennzeichnet, daß das Applet auf einer Webseite mindestens einer der ersten oder zweiten Systemeinheiten ausgeführt wird.
20. Verfahren nach einem der Ansprüche 15 bis 19, dadurch gekennzeichnet, daß mindestens eine der Funktionseinheiten als JAVA Applet ausgelegt wird.
21. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die Ausgabeinformationen als grafische Benutzeroberfläche der Anwendung repräsentiert werden.

22. Verfahren nach Anspruch 21, dadurch gekennzeichnet, daß die Benutzeroberfläche im Bitmap- oder GIF-Format repräsentiert wird.
23. Verfahren nach einem der Ansprüche 13 oder 15 bis 22, dadurch gekennzeichnet, daß die Ausgabeinformationen mittels eines bei mindestens einer der Systemeinheiten vorliegenden Web-Browsers präsentiert werden.
24. Verfahren nach einem der Ansprüche 13 oder 15 bis 23, dadurch gekennzeichnet, daß eine gemeinsame Bearbeitung einer Anwendung bei der dritten Systemeinheit vorab angemeldet wird. 5

Hierzu 5 Seite(n) Zeichnungen

10

15

20

25

30

35

40

45

50

55

60

65

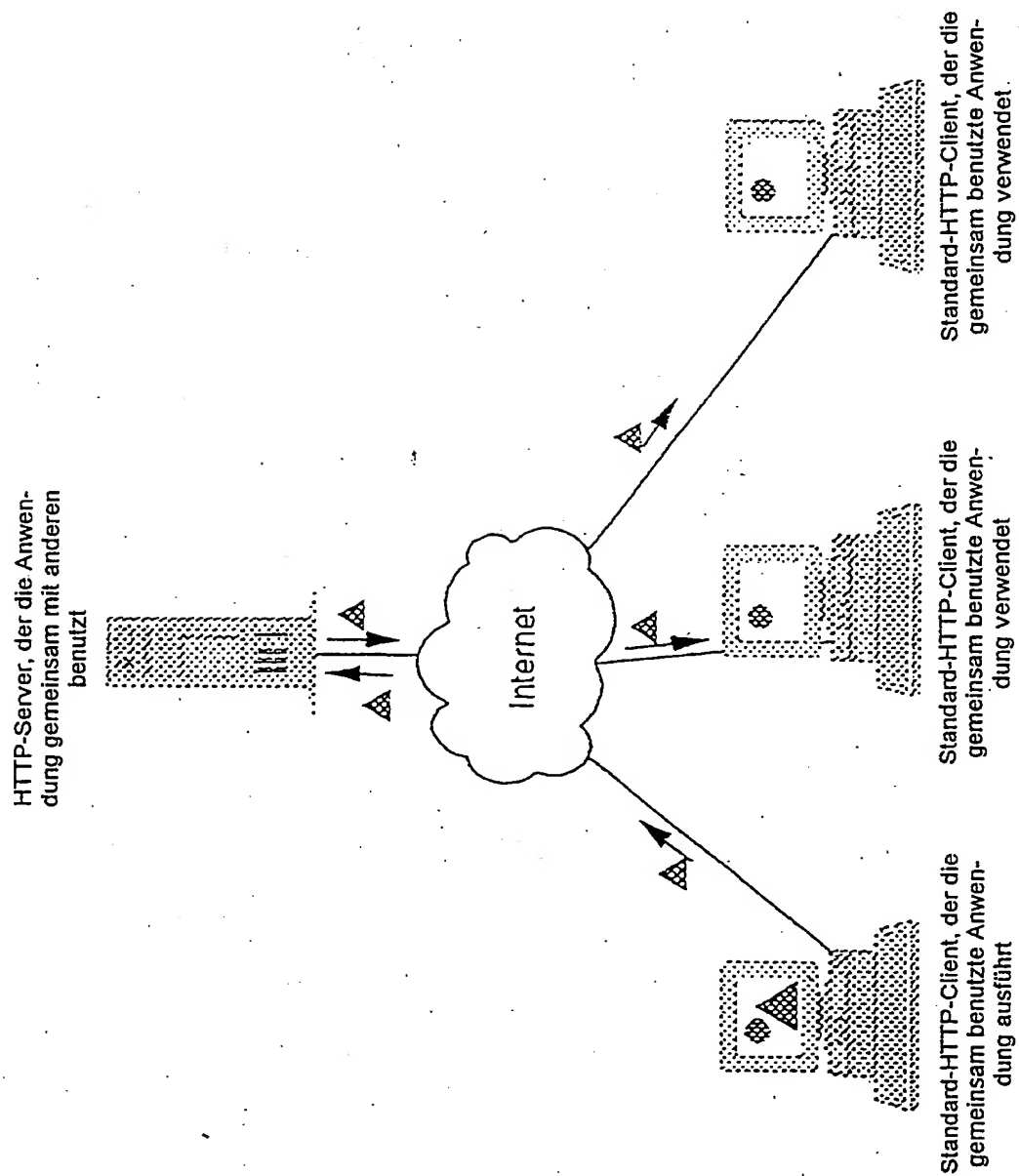
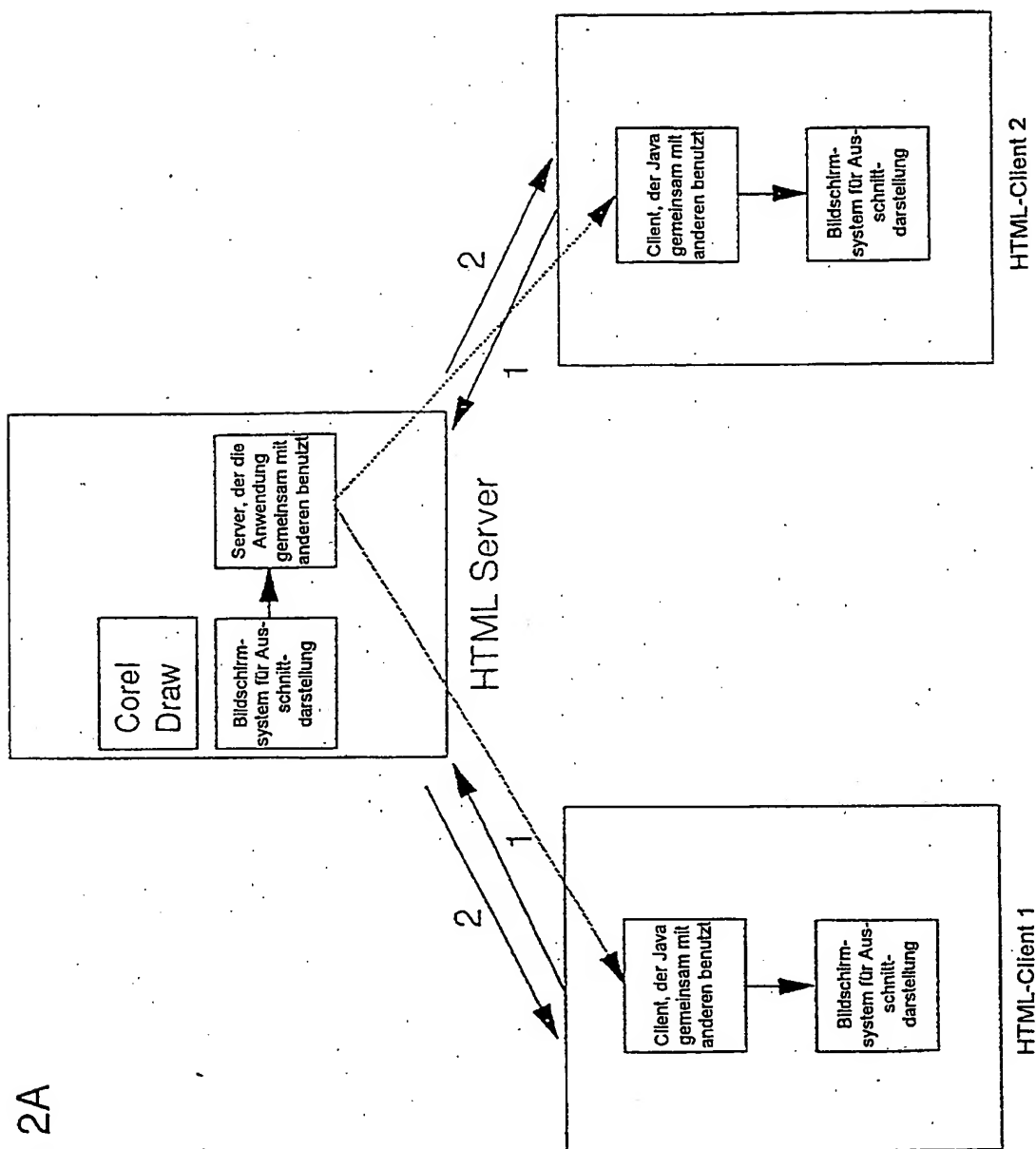
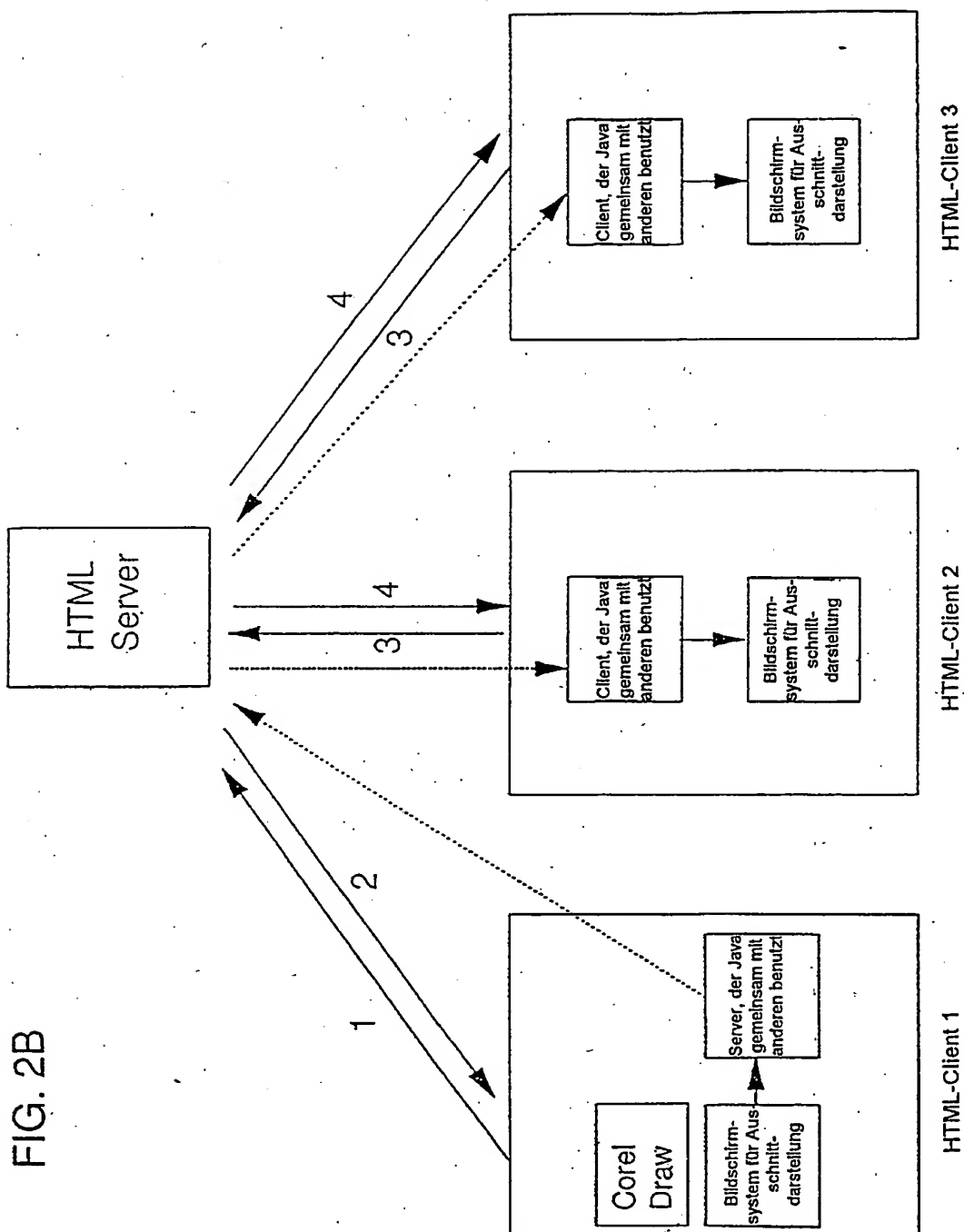


FIG. 1





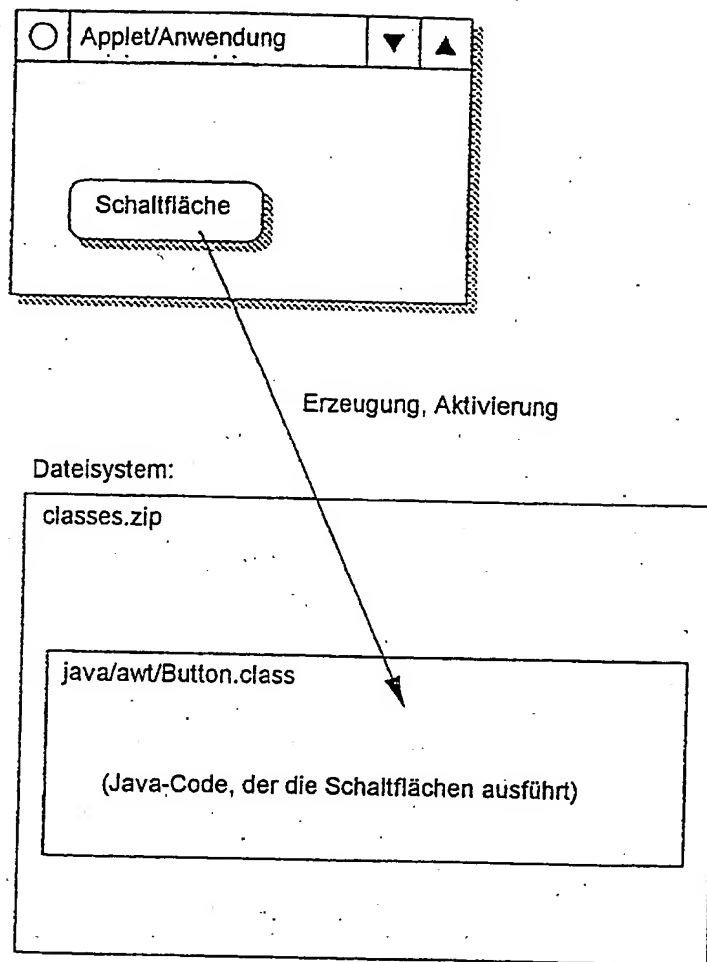


FIG. 3

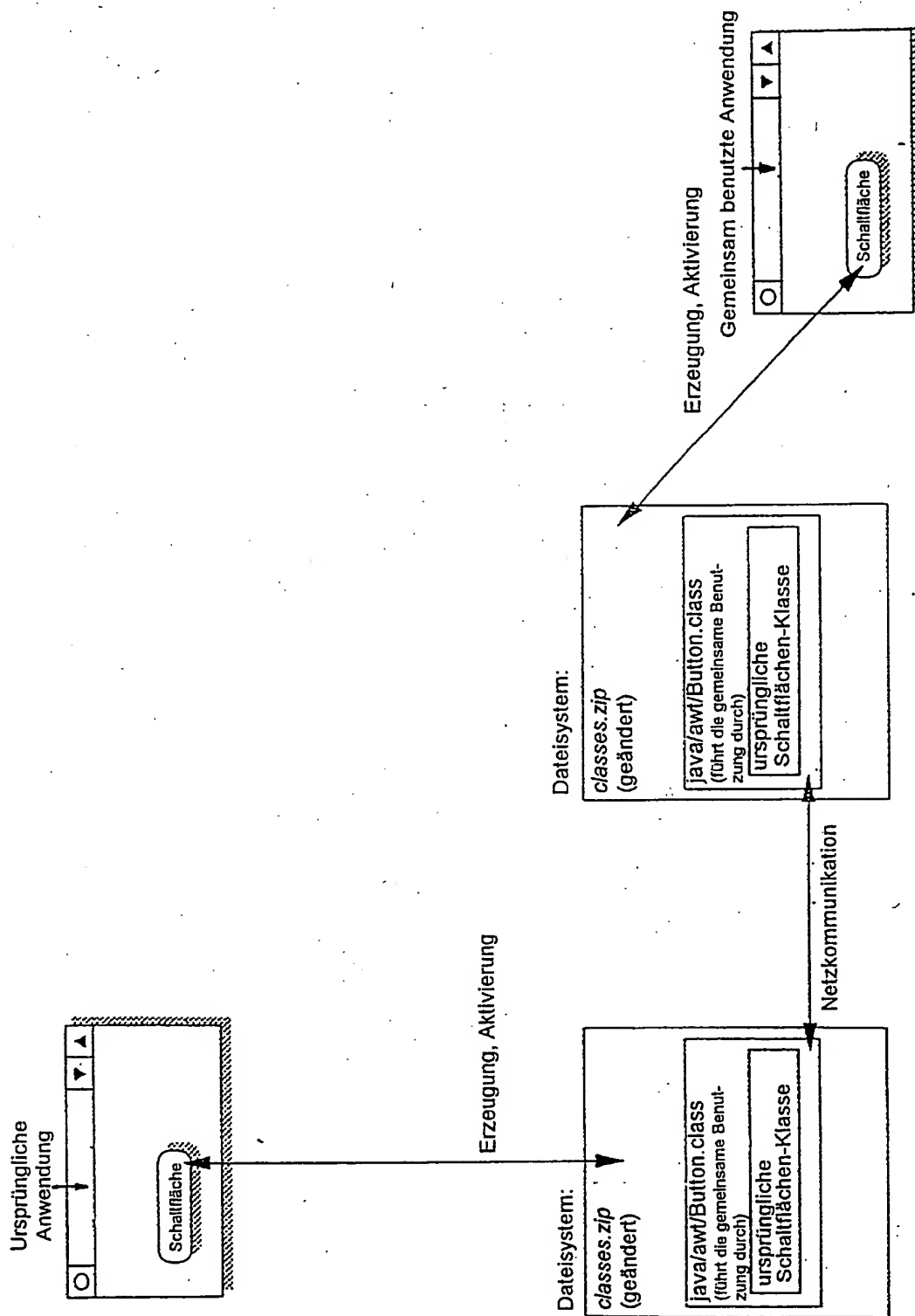


FIG. 4

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.